CLOCKS
Clocks in Digital Systems

• Why are clocks and clocked memory registers needed inside digital systems?
• Clocks pace the flow of data inside digital processors
• The exact speed of data through circuits is impossible to predict accurately due to factors such as:
  – Fabrication process variations
  – Supply voltage variations
  – Temperature variations
  – Countless parasitic effects (e.g., wire-to-wire capacitances)
  – Data-dependent variations (e.g., calculating $1 \text{ OR } 1 = 1$ requires a different delay than $1 \text{ OR } 0 = 1$)

© B. Baas
Clocks in Digital Systems

• Clocked memory elements slow down the fastest signals, wait until *all* signals have finished propagating through the combinational logic in the stage*, and then release them into the next stage simultaneously, controlled by the active edge of the clock signal.

• * This is why we care about only the single slowest signal in a block (max propagation delay) when finding the maximum clock frequency.
Clocks in Digital Systems

• All paths within a digital system consist of an input register, (optionally) followed by combinational logic, followed by an output register

• Therefore:
  – If we can make this structure work under all conditions, we can build a robust digital system
  – We should analyze this structure carefully

© B. Baas
Robust Clock Design

- Edge-triggered memory elements (flip-flops) are generally more robust than level-sensitive memory elements (transparent latches)
- Always follow these rules in this class, and for the most robust designs:
  1. Only clock signals may connect to flip-flop or latch clock inputs
     - A simpler circuit may sometimes be possible if a logic signal is connected to a clock input, but do not do it for robustness
     - `always @(posedge key) begin`
  2. Clock signals may not connect to any node other than a flip-flop or latch clock input
     - No logic gate inputs
     - No flip-flop or latch inputs other than the clock input
Robust Clock Design

- There are a few common exceptions to these rules—but they are topics outside the scope of this class. For example:
  - Clock generation circuits
  - Clock gating circuits
  - Clock tree buffer circuits

- These are circuit-level issues that must be designed carefully using *spice*